

A polynomial time computable metric between point sets

Jan Ramon and Maurice Bruynooghe

Department of Computer Science
Katholieke Universiteit Leuven
Celestijnenlaan 200A, B-3001 Heverlee, Belgium
email: {Jan.Ramon,Maurice.Bruynooghe}@cs.kuleuven.ac.be

The date of receipt and acceptance will be inserted by the editor

Abstract. Measuring the similarity or distance between two sets of points in a metric space is an important problem in machine learning and has also applications in other disciplines e.g. in computational geometry, philosophy of science, methods for updating or changing theories, Recently Eiter and Mannila have proposed a new measure which is computable in polynomial time. However, it is not a distance function in the mathematical sense because it does not satisfy the triangle inequality.

We introduce a new measure which is a metric while being computable in polynomial time. We also present a variant which computes a normalised metric and a variant which can associate different weights with the points in the set.

1 Introduction

In many applications it is desirable to measure the similarity or difference between objects i.e. to express it by a single numeral. Ideally such a measure has the properties of a metric:

Definition 1 (metric). *Given a nonempty set of objects O , a metric d is a mapping $O \times O \rightarrow \mathbb{R}^+$ such that for all $x, y, z \in O$:*

1. $d(x, y) = 0 \Leftrightarrow x = y$,
2. $d(x, y) = d(y, x)$ (*symmetry*),
3. $d(x, z) \leq d(x, y) + d(y, z)$ (*triangle inequality*).

In the sequel, measures which satisfy only the first two properties are called similarity measures.

The problem we study is the following: given some set X and a metric d on X , how can we extend d into a metric on the set of all (finite) subsets of X .

Distances between composed objects and between sets of objects have applications in many domains such as cluster analysis (e.g. TIC [2], KBG [1]), computational geometry [8], machine learning (e.g. [9, ch.4], RIBL [6]),

Existing proposals for measures between point sets all have some problems: some are trivial and not very well suited for applications (e.g. the Hausdorff metric), others do not satisfy all the properties of metrics (e.g. the similarity measures in [5]).

In this paper we present a measure between point sets which is a metric while avoiding the drawbacks of the Hausdorff metric. We show that it is computable in polynomial time.

Some elementary notions about binary relations and basic definitions about flow networks are recalled in section 2. The latter will be used to prove that our metric is computable in polynomial time. The Hausdorff metric and the similarity measures discussed by Eiter and Mannila [5] are reviewed in section 3. Some of the latter are concisely presented as instances of a novel general schema. In section 4, we introduce another instance of this general schema and prove that it is a metric (satisfying the triangle inequality) and computable in polynomial time. We develop a normalised metric in section 5. A generalisation of the metric which associates weights with the points in the set and which is better suited to measure the distance between sets of very different sizes is developed in section 6. In section 7 some applications from the machine learning area are discussed. We end with a brief summary in section 8.

This paper is an extension of some of the material in [11].

2 Preliminaries

Let $\#S$ denote the cardinality of a set S ; $|n|$ denotes the absolute value of a number n ; for a relation $f \subseteq A \times B$, $f(x)$ denotes the set $\{y \mid (x, y) \in f\}$, $f(S)$ denotes the set $\{y \mid \exists x \in S \wedge (x, y) \in f\}$, $\#f(A)$ is abbreviated as $\#f$ and f^{-1} denotes the relation $\{(y, x) \mid (x, y) \in f\}$.

Definition 2. *A relation $f \subseteq A \times B$ between two finite sets A and B is a surjection from A onto B if $\forall (a, b), (c, d) \in f : (a = c \Rightarrow b = d)$ and $\forall b \in B, \exists a \in A : (a, b) \in f$. A surjection f from A onto B*

is fair if $\forall x, y \in B : |\#(f^{-1}\{x\}) - \#(f^{-1}\{y\})| \leq 1$, so f maps the elements of A on elements of B as evenly as possible. A linking $f \subseteq A \times B$ is a relation such that $\forall a \in A, \exists b \in B : (a, b) \in f$ and $\forall b \in B, \exists a \in A : (a, b) \in f$, so all elements of A are associated with at least one of B and vice versa. A matching f between A and B is a relation such that $\forall (a, b), (c, d) \in f : (a = c \Leftrightarrow b = d)$, so each element of A is associated with at most one element of B and vice versa. A matching f between A and B is **maximal** if there is no $(a, b) \in A \times B \setminus f$ such that $f \cup \{(a, b)\}$ is a matching between A and B . A perfect matching is a maximal matching between two sets of equal cardinality.

Finally, we recall some definitions on transport networks from [13].

Definition 3 (indegree and outdegree). If (V, E) is a directed graph and $v \in V$, then $\text{deg}_{in}(v) = \#\{x \in V | (x, v) \in E\}$ and $\text{deg}_{out}(v) = \#\{x \in V | (v, x) \in E\}$.

Definition 4 (transport network). $N(V, E, \text{cap}, s, t)$ is called a transport network iff (V, E) is a loop-free connected finite directed graph with $s, t \in V$, $\text{deg}_{in}(s) = 0$, $\text{deg}_{out}(t) = 0$ and cap is a function $\text{cap} : E \rightarrow \mathbb{R}^+$.

Definition 5 (weighted transport network). $N(V, E, \text{cap}, s, t, w)$ is called a weighted transport network iff $N(V, E, \text{cap}, s, t)$ is a transport network and w is a function $w : E \rightarrow \mathbb{R}^+$.

Definition 6 (flow). If $N(V, E, \text{cap}, s, t, w)$ is a weighted transport network, then a function f from E to \mathbb{R} is a flow for N iff

- $\forall e \in E : f(e) \leq \text{cap}(e)$.
- $\forall v \in V \setminus \{s, t\} : \sum_{u \in V} f(v, u) = \sum_{u \in V} f(u, v)$ (if there is no edge $(v, u) \in E$, then $f(v, u) = 0$). This is called the continuity property.

Definition 7 (value of a flow). If f is a flow for $N(V, E, \text{cap}, s, t, w)$, then $\text{val}(f) = \sum_{v \in V} f(s, v) = \sum_{v \in V} f(v, t)$ is called the value of f .

Definition 8 (weight of a flow). If f is a flow for $N(V, E, \text{cap}, s, t, w)$, then the weight of f is $w(f) = \sum_{e \in E} w(e) \cdot f(e)$.

Definition 9 (maximal flow minimal weight flow). If f is a flow for $N(V, E, \text{cap}, s, t, w)$, then f is called a maximal flow if for all flows f' for N , $\text{val}(f') \leq \text{val}(f)$ and f is called a maximal flow minimal weight flow iff for all maximal flows f' for N , $w(f') \geq w(f)$.

Definition 10 (integer flow). *If f is a flow for $N(V, E, cap, s, t, w)$, then f is called an integer flow iff for all edges $(a, b) \in E$, $f(a, b)$ is an integer.*

Definition 11 (integer flow network). *If $N(V, E, cap, s, t, w)$ is a weighted transport network, then N is called an integer flow network iff for all edges $(a, b) \in E$, $cap(a, b)$ is an integer.*

In [10] the following theorem is proved:

Theorem 1. *If $N(V, E, cap, s, t, w)$ is an integer flow network then there is a maximal flow minimal weight flow f for N such that f is an integer flow.*

3 Distances between sets of points

In this section we discuss some existing distance measures between sets of points.

The Hausdorff metric Well known is the Hausdorff metric. Given X , a set of points, and d , a metric between points, $d_h : 2^X \times 2^X \rightarrow \mathbb{R}$ is defined as:

$$d_h(A, B) = \max \left(\max_{a \in A} (\min \{d(a, b) | b \in B\}), \max_{b \in B} (\min \{d(a, b) | a \in A\}) \right)$$

While this function has all the properties of a metric, it does not take into account much information about the points in the sets (it is determined by the distance of the most distant element of both sets to the nearest neighbour in the other set). This makes this metric unsuited for applications where one set has likely a point which is very different from all points of the other set as e.g. in Inductive Logic Programming [11].

Sum of minimal distance measure Eiter and Mannila [5] discuss the sum of minimal distances similarity measure. It is defined as:

$$d(X, Y) = \frac{1}{2} \left(\sum_{x \in X} (\min_{y \in Y} d(x, y)) + \sum_{y \in Y} (\min_{x \in X} d(x, y)) \right)$$

However, this is in general not a metric.

Distances based on optimal mappings. Eiter and Mannila [5] also discuss a family of Manhattan measures between sets which we can describe as instances of the following scheme:

$$d^\beta(A, B) = \min_{r \in m^\beta(A, B)} d(r, A, B)$$

where

$$d(r, A, B) = \left[\sum_{(x, y) \in r} d(x, y) \right] + \frac{\#(B \setminus r(A)) + \#(A \setminus r^{-1}(B))}{2} \cdot M$$

In this formula, m^β is a function that maps each pair $(A, B) \in 2^X \times 2^X$ to a relation between A and B (a subset of $A \times B$) and M is a constant (representing a large or the maximal possible distance between 2 points).

This means that one sums the distances of the pairs of elements which are in r and adds a penalty $M/2$ for each element that does not match with an element from the other set.

The authors discuss three instantiations:

- $m^\beta = m^s$ with $m^s(A, B)$ the set of all surjections from the larger of A and B to the smaller of A and B (surjection-measure d^s).
- $m^\beta = m^{fs}$ with $m^{fs}(A, B)$ the set of all fair surjections from the larger of A and B to the smaller of A and B (fair surjection-measure d^{fs}).
- $m^\beta = m^l$ with $m^l(A, B)$ the set of all linkings between A and B (linking-measure d^l).

They show that these similarity measures can be evaluated in polynomial time. They are not metrics as the triangle inequality is violated. Note that d^β agrees with d on singletons: $\forall x, y \in X : d^\beta(\{a\}, \{b\}) = d(a, b)$.

4 A metric based on optimal matchings

Using matchings ($m^\beta = m^m$ with $m^m(A, B)$ the set of all matchings between A and B) instead of surjections, fair surjections or linkings, one obtains another instantiation of the schema presented in the previous section:

$$d^m(A, B) = \min_{r \in m^m(A, B)} d(r, A, B)$$

Definition 12. *A matching r is optimal for the distance between A and B iff $d^m(A, B) = d(r, A, B)$.*

The following two theorems are a corollary of theorems 7 and 8 in section 6.

Theorem 2. d^m is a metric for $M > 0$.

Theorem 3. If the time to compute the distance between two points is bounded by T , then the time to compute $d^m(A, B)$ is bounded by a polynomial in $\#A$, $\#B$ and T .

The interested reader can find a direct proof of these theorems in [12].

5 Normalised matching metric.

In some applications (e.g. algorithms where the distance between clusters shouldn't depend on the size of the objects) it is desirable to work with normalised distances i.e. distances in the interval $[0, 1]$. Instance based learning systems such as RIBL [7] and clustering algorithms (e.g. agglomerative clustering algorithms using distances). make use of normalised similarity measures. Also here, a normalised metric is preferable above a normalised similarity which is not a metric. In this section we show how a normalised metric between points can be extended to a normalised metric between point sets.

With normalisation, the maximal distance between two points is 1, so M can be set to 1 and the general formula for distances between sets can be simplified into:

$$\begin{aligned} d^m(A, B) &= \sum_{(x,y) \in m_{AB}} d(x, y) + \frac{\#(B \setminus m_{AB}(A)) + \#(A \setminus m_{AB}^{-1}(B))}{2} \\ &= \sum_{(x,y) \in m_{AB}} d(x, y) + \frac{\#B + \#A - 2\#m_{AB}}{2} \end{aligned} \tag{1}$$

where m_{AB} is an optimal matching for $d^m(A, B)$.

We define:

$$\begin{aligned} d^{m,n}(A, B) &= \text{if } A = \emptyset \text{ and } B = \emptyset \text{ then } 0 \\ &\text{else } \frac{2 \cdot d^m(A, B)}{d^m(A, B) + (\#A + \#B)/2} \end{aligned} \tag{2}$$

Note that $d^{m,n}$ is normalised. Indeed, $0 \leq d^m(A, B) \leq (\#A + \#B)/2$, hence $2 \cdot d^m(A, B) \leq d^m(A, B) + (\#A + \#B)/2$.

Before proving that $d^{m,n}$ is a metric, we introduce size functions for point sets.

Definition 13 (size). Let U be a universe. A size is a function $s : 2^U \rightarrow \mathbb{R}$ such that $\forall X \in 2^U : s(X) \geq 0$ and $\forall X, Y \in 2^U : (X \cap Y = \emptyset) \Rightarrow s(X \cup Y) = s(X) + s(Y)$.

In the sequel we consider sets in some universe U and a size function s over U and use the notion of symmetric difference: $A\Delta B = (A \setminus B) \cup (B \setminus A)$.

Lemma 1.

$$s(A \cup B) = [s(A) + s(B) + s(A\Delta B)]/2$$

PROOF: Applying the definitions we get $s(A \cup B) = s(A) + s(B \setminus A) = s(B) + s(A \setminus B)$. Hence $2s(A \cup B) = s(A) + s(B \setminus A) + s(B) + s(A \setminus B) = s(A\Delta B) + s(A) + s(B)$. 2

Definition 14. $\Delta_{s,n}(A, B) =$ if $(A \cup B = \emptyset)$ then 0 else $\frac{s(A\Delta B)}{s(A \cup B)}$.

Theorem 4. $\Delta_{s,n}$ is a normalised metric on 2^U .

PROOF:

$0 \leq s(A\Delta B) \leq s(A \cup B)$ hence $0 \leq \Delta_{s,n}(A, B) \leq 1$. The other properties are trivial except the triangle inequality in case that A , B and C are non-empty.

$$\begin{aligned} & \Delta_{s,n}(A, B) + \Delta_{s,n}(B, C) \\ &= \frac{s(A\Delta B)}{s(A \cup B)} + \frac{s(B\Delta C)}{s(B \cup C)} \\ &= \frac{s((A \cap C) \setminus B) + s((A \setminus C) \setminus B) + s((B \cap C) \setminus A) + s((B \setminus C) \setminus A)}{s(A \setminus C) + s(B \cap C) + s((A \cap C) \setminus B) + s((B \setminus C) \setminus A)} \\ & \quad + \frac{s((A \cap C) \setminus B) + s((C \setminus A) \setminus B) + s((B \cap A) \setminus C) + s((B \setminus C) \setminus A)}{s(C \setminus A) + s(B \cap A) + s((A \cap C) \setminus B) + s((B \setminus C) \setminus A)} \\ & \geq \frac{s((A \setminus C) \setminus B) + s((B \cap C) \setminus A)}{s(A \setminus C) + s(B \cap C)} + \frac{s((C \setminus A) \setminus B) + s((B \cap A) \setminus C)}{s(C \setminus A) + s(B \cap A)} \\ & \geq \frac{s((A \setminus C) \setminus B) + s((B \cap C) \setminus A)}{s(A \cup C)} + \frac{s((C \setminus A) \setminus B) + s((B \cap A) \setminus C)}{s(A \cup C)} \\ & \geq \frac{s(A\Delta C)}{s(A \cup C)} \\ & \geq \Delta_{s,n}(A, C) \end{aligned}$$

Lemma 2. *Let X and Y be point sets. If sets A and B and size s exist such that $s(A) = \#X$, $s(B) = \#Y$ and $s(A\Delta B) = 2d^m(X, Y)$, then $d^{m,n}(X, Y) = \frac{s(A\Delta B)}{s(A\cup B)} = \Delta_{s,n}(A, B)$.*

PROOF:

$$\text{We have } d^{m,n}(X, Y) = \frac{2d^m(X, Y)}{(\#X + \#Y)/2 + d^m(X, Y)} = \frac{s(A\Delta B)}{[s(A) + s(B) + s(A\Delta B)]/2}.$$

From lemma 1, there follows $d^{m,n}(X, Y) = \frac{s(A\Delta B)}{s(A\cup B)} = \Delta_{s,n}(A, B)$. 2

Note that there does not exist a size such that $s(X) = \#X$, $s(Y) = \#Y$ and $s(X\Delta Y) = 2d^m(X, Y)$, since it is possible that $X \cap Y = \emptyset$, hence $s(X\Delta Y) = \#X + \#Y$ and at the same time $2d^m(X, Y) < \#X + \#Y$. This is the motivation for introducing sets A and B . The following lemma shows that sets with the desired sizes exist.

Lemma 3. *Let X_i ($i = 0, 1, 2$ indices modulo 3) be point sets. There exists a universe U , a size $s : 2^U \rightarrow \mathbb{R}^+$ and sets $A_i \subset U$ such that for all i, j , $s(A_i) = \#X_i$ and $s(A_i\Delta A_j) = 2d^m(X_i, X_j)$.*

PROOF: It suffices to show that s can assign non-negative values to the elementary sets $A_i \cap A_{i+1} \cap A_{i+2}$, $(A_i \cap A_{i+1}) \setminus A_{i+2}$ and $(A_i \setminus A_{i+1}) \setminus A_{i+2}$, from which the sets A_i are composed such that the sizes given in the lemma are satisfied.

Let $x_i = \#X_i$ and $d_{i,j} = 2d^m(X_i, X_j)$. Then, from equation 1 at the beginning of this section it follows that

$$|x_i - x_j| \leq d_{i,j} \leq x_i + x_j. \quad (3)$$

Let $y_i = \frac{1}{2}(x_i + x_{i+1} - d_{i,i+1})$. These y_i are nonnegative by equation (3). Without loss of generality we can assume that the indices are assigned such that $y_0 = \min\{y_0, y_1, y_2\}$. Assign

$$s(A_i \cap A_{i+1} \cap A_{i+2}) = y_0$$

$$s((A_i \cap A_{i+1}) \setminus A_{i+2}) = y_i - y_0$$

These values are all nonnegative. It follows that $s(A_i \cap A_{i+1}) = s(A_i \cap A_{i+1} \cap A_{i+2}) + s((A_i \cap A_{i+1}) \setminus A_{i+2}) = y_i$. To ensure that $s(A_i) = x_i$ we assign

$$\begin{aligned} s((A_i \setminus A_{i+1}) \setminus A_{i+2}) &= x_i - s(A_i \cap A_{i+1}) - s((A_{i+2} \cap A_i) \setminus A_{i+1}) \\ &= x_i - y_i - (y_{i+2} - y_0) \\ &= \frac{1}{2}(x_0 + x_1 - d_{0,1} - x_{i+1} + d_{i,i+1} - x_{i+2} + d_{i+2,i}) \end{aligned}$$

These values should be nonnegative for $i = 0, 1, 2$.

$$i = 0 : s((A_0 \setminus A_1) \setminus A_2) = \frac{1}{2}(x_0 - x_2 + d_{2,0}) \quad (\text{by (3)})$$

$$i = 1 : s((A_1 \setminus A_2) \setminus A_0) = \frac{1}{2}(x_1 - x_2 + d_{1,2}) \quad (\text{by (3)})$$

$$i = 2 : s((A_2 \setminus A_0) \setminus A_1) = \frac{1}{2}(-d_{0,1} + d_{2,0} + d_{1,2}) \quad (d^m \text{ is a metric})$$

Finally, we know that $s(A_i \Delta A_{i+1}) = s(A_i) + s(A_{i+1}) - 2s(A_i \cap A_{i+1}) = x_i + x_{i+1} - 2y_i = d_{i,i+1}$. This completes the proof. \square

Theorem 5. $d^{m,n}$ is a normalised metric.

PROOF:

Let X_i ($i = 0, 1, 2$) be sets. By lemma 3, there exists a size s and sets A_i ($i = 0, 1, 2$) such that $s(A_i) = \#X_i$ and $s(A_i \Delta A_j) = 2 \cdot d^m(X_i, X_j)$ for $i, j \in \{0, 1, 2\}$. Hence lemma 2 is applicable, i.e. $d^{m,n}(X_i, X_j) = \Delta_{s,n}(A_i, A_j)$. By theorem 4, $\Delta_{s,n}$ is a normalised metric, hence also $d^{m,n}$ is a normalised metric. \square

Theorem 6. If $d(a, b)$ is computable in polynomial time, then $d^{m,n}(A, B)$ is computable in polynomial time.

PROOF:

This follows from the fact that $d^m(A, B)$ is computable in polynomial time and that $d^{m,n}(A, B)$ can be computed in constant time from $d^m(A, B)$.

6 Generalisation

A weakness of the measures presented so far is that the distance between a large and a small set is largely determined by their difference in cardinality (if $\#A \gg \#B$, then $d(A, B) \approx (\#A - \#B) \cdot M/2$). By associating appropriate weights with the elements in the sets, it is possible to generalize the notion of cardinality in a way such that sets of vastly different cardinality can be scaled appropriately. Weights could also be used to give different importance to the members of a set.

Definition 15 (Weighting function). A function $W : 2^X \rightarrow (X \rightarrow \mathbb{R}^+)$ is a weighting function for X .

Definition 16 (Size under weighting function). Let W be a weighting function for X . Then the function $size_W : 2^X \rightarrow \mathbb{R}^+$ is defined as $size_W(A) = \sum_{a \in A} W[A](a)$.

Example 1. The function W_1 with $W_1[A] = \{(a, 1) | a \in A\}$ is a weighting function such that for all objects $A \in 2^X$, $size_{W_1}(A) = \#A$.

Definition 17. Let X be a set and let W be a weighting function for X . We define $Q_X^W = \max_{A \in 2^X} size_W(A)$.

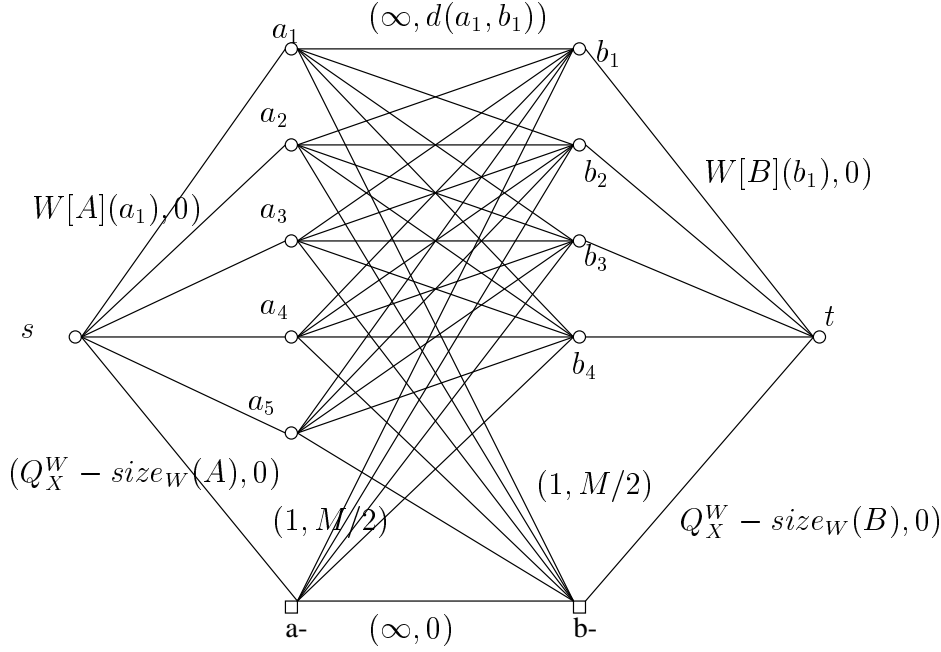


Fig. 1 A distance network

Definition 18 (distance network). Let X be a set, and d a metric on X . Let M be a constant and let W be a weighting function for X . Then for all finite $A, B \in 2^X$ with $A = \{a_1, \dots, a_m\}$ and $B = \{b_1, \dots, b_n\}$, we define a distance network between A and B for d, M and W in X to be $N[X, d, M, W, A, B] = N(V, E, \text{cap}, s, t, w)$ with $V = A \cup B \cup \{s, t, a_-, b_-\}$, $E = (\{s\} \times (A \cup \{a_-\})) \cup ((B \cup \{b_-\}) \times \{t\}) \cup ((A \cup \{a_-\}) \times (B \cup \{b_-\}))$, $\forall a \in A, \forall b \in B : w(s, a) = w(b, t) = w(s, a_-) = w(b_-, t) = w(a_-, b_-) = 0 \wedge w(a, b) = d(a, b) \wedge w(a_-, b) = w(a, b_-) = M/2$ and $\forall a \in A, \forall b \in B : \text{cap}(s, a) = W[A](a) \wedge \text{cap}(b, t) = W[B](b) \wedge \text{cap}(s, a_-) = Q_X^W - \text{size}_W(A) \wedge \text{cap}(b_-, t) = Q_X^W - \text{size}_W(B) \wedge \text{cap}(a, b) = \text{cap}(a_-, b) = \text{cap}(a, b_-) = \text{cap}(a_-, b_-) = \infty$.

Note that the definition of Q_X^W ensures $\text{cap}(s, a_-) \geq 0$ and $\text{cap}(b_-, t) \geq 0$. Moreover, $\text{cap}(s, a_-) + \sum_{i=1}^m \text{cap}(s, a_i) = \text{cap}(b_-, t) + \sum_{i=1}^n \text{cap}(b_i, t) = Q_X^W$. Hence:

Proposition 1. The flow of a maximal flow minimal weight flow of a distance network is Q_X^W .

Definition 19 (netflow distance). Let X be a set, d a metric on X , M a constant, and W a weighting function for X . For all $A, B \in$

2^X , the netflow distance from A to B under d , M and W in X , denoted $d_{X,d,M,W}^N(A, B)$, is the weight of the minimal weight maximal flow from s to t in $N[X, d, M, W, A, B]$.

Proposition 2. *With all the weights set to 1, the netflow distance $d_{X,d,M,W}^N(A, B)$ from A to B is equal to the distance from A to B under the optimal matching $d^m(A, B)$.*

PROOF:

We have $\forall a \in A : W[A](a) = 1$ and $\forall b \in B : W[B](b) = 1$.

Given a matching r , one can define a flow f as follows:

$$\begin{aligned} f(a_i, b_-) &= \text{if } \exists b_j \in B : (a_i, b_j) \in r \text{ then } 0 \text{ else } 1 \\ f(a_-, b_i) &= \text{if } \exists a_i \in A : (a_i, b_j) \in r \text{ then } 0 \text{ else } 1 \\ f(a_i, b_j) &= \text{if } (a_i, b_j) \in r \text{ then } 1 \text{ else } 0 \end{aligned}$$

One can easily verify that f is a maximal flow and that its weight is equal to $d(r, A, B) = d^m(A, B)$. Hence $d^m(A, B) \leq d_{X,d,M,W}^N(A, B)$

Conversely, since we have an integer flow network, there exists an integer flow f which is maximal and has minimal weight by theorem 1. It can be easily verified that there is a corresponding matching r such that $d(r, A, B)$ is equal to the weight of this flow. Hence $d^m(A, B) \geq d_{X,d,M,W}^N(A, B)$. 2

Notation 1 We use $\sum_{i \in \{1..m, -\}} expr_i$ as abbreviation of $\sum_{i=1}^m expr_i + expr_-$ and d^N as abbreviation of $d_{X,d,M,W}^N$.

Theorem 7. *The netflow distance is a metric.*

PROOF: Using the notations of definition 18, we prove:

– $d^N(A, A) = 0$.

If $B = A$ then $a_i = b_i$ and $d(a_i, b_i) = 0$. With 0-weight flows from s to a_i to b_i to t and from s to a_- to b_- to t , a 0-weight solution is obtained so $d^N(A, A) = 0$.

– $d^N(A, B) = d^N(B, A)$.

This follows from the fact that the solution of a minimal weight maximal flow problem has the same weight as the solution obtained when the source and sink are reversed.

– $d^N(A, B) + d^N(B, C) \geq d^N(A, C)$

Figure 2 is the distance network between A and B followed by the distance network between B and C . For the latter, the nodes have been renamed into b'_1, \dots, b'_l . $d^N(A, B) + d^N(B, C)$ is the weight of the solution of the minimal weight maximal flow problem in

this figure. Similarly, $d^N(A, C)$ is the weight of the solution of the minimal weight maximal flow problem in figure 3.

We now prove that for each flow f_1 in the network of figure 2, there exists a flow f_2 in the network of figure 3 that has the same total flow and has a smaller or equal weight.

Consider a flow f_1 in figure 2. Then, let

$$- f_2(s^*, a_i^*) = f_1(s, a_i), f_2(s^*, a_-^*) = f_1(s, a_-), f_2(c_i^*, t^*) = f_1(c_i, t) \\ \text{and } f_2(c_-^*, t^*) = f_1(c_-, t).$$

$$- f_2(a_i^*, c_k^*) = \sum_{j \in \{1..n, -\}} \frac{f_1(a_i, b_j) \cdot f_1(b'_j, c_k)}{TB_j} \text{ where} \\ TB_j = \sum_{i \in \{1..m, -\}} f_1(a_i, b_j)$$

First we prove a property of TB_j :

$$\begin{aligned} TB_j &= \sum_{i \in \{1..m, -\}} f_1(a_i, b_j) \\ &= f_1(b_j, r) && \text{(continuity in } b_j) \\ &= \text{cap}(b_j, r) && \text{(saturated to reach flow } Q_X^W) \\ &= \text{cap}(r, b'_j) && \text{(by construction)} \\ &= f_1(r, b'_j) && \text{(saturated to reach flow } Q_X^W) \\ &= \sum_{k \in \{1..r, -\}} f_1(b'_j, c_k) && \text{(continuity in } b'_j) \end{aligned}$$

Next we verify that f_2 is a flow for the network in figure 3. The non-trivial part is the continuity in a_i^* and c_k^* . In a_i^* :

$$\begin{aligned} &\sum_u f_2(u, a_i^*) - \sum_u f_2(a_i^*, u) \\ &= f_2(s^*, a_i^*) - \sum_{k \in \{1..r, -\}} f_2(a_i^*, c_k^*) \\ &= f_1(s, a_i) - \sum_{k \in \{1..r, -\}} \sum_{j \in \{1..n, -\}} \frac{f_1(a_i, b_j) \cdot f_1(b'_j, c_k)}{TB_j} \\ &= f_1(s, a_i) - \sum_{j \in \{1..n, -\}} \frac{f_1(a_i, b_j) \cdot \sum_{k \in \{1..r, -\}} f_1(b'_j, c_k)}{TB_j} \\ &= f_1(s, a_i) - \sum_{j \in \{1..n, -\}} f_1(a_i, b_j) \\ &= 0 \end{aligned}$$

The continuity in c_k^* can be verified similarly.

The value of f_2 is Q_X^W , hence f_2 is a maximal flow. Finally we show that f_2 has a weight smaller or equal to the weight of f_1 . We have

$$w(f_2) = \sum_{i \in \{1..m, -\}} \sum_{k \in \{1..r, -\}} f_2(a_i^*, c_k^*) w(a_i^*, c_k^*)$$

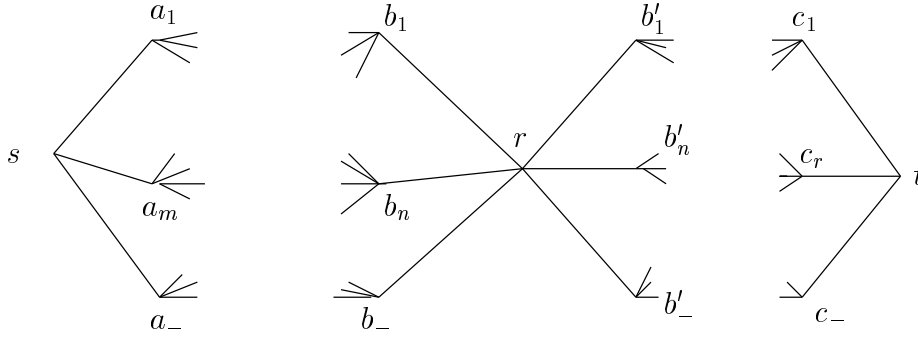


Fig. 2 Network for $d^N(A, B) + d^N(B, C)$

$$\begin{aligned}
&= \sum_{i \in \{1..m, -\}} \sum_{k \in \{1..r, -\}} \sum_{j \in \{1..n, -\}} \frac{f_1(a_i, b_j) \cdot f_1(b'_j, c_k)}{TB_j} w(a_i^*, c_k^*) \\
&\quad (\text{since } w(a_i^*, c_k^*) = d(a_i^*, c_k^*) \leq d(a_i, b_j) + d(b'_j, c_k)) \\
&= w(a_i, b_j) + w(b'_j, c_k) \\
&\leq \sum_{i \in \dots} \sum_{k \in \dots} \sum_{j \in \dots} \frac{f_1(a_i, b_j) \cdot f_1(b'_j, c_k)}{TB_j} [w(a_i, b_j) + w(b'_j, c_k)] \\
&= \sum_{i \in \{1..m, -\}} \sum_{k \in \{1..r, -\}} \sum_{j \in \{1..n, -\}} \frac{f_1(a_i, b_j) \cdot f_1(b'_j, c_k)}{TB_j} w(a_i, b_j) \\
&\quad + \sum_{i \in \{1..m, -\}} \sum_{k \in \{1..r, -\}} \sum_{j \in \{1..n, -\}} \frac{f_1(a_i, b_j) \cdot f_1(b'_j, c_k)}{TB_j} w(b'_j, c_k) \\
&= \sum_{i \in \{1..m, -\}} \sum_{j \in \{1..n, -\}} f_1(a_i, b_j) w(a_i, b_j) \\
&\quad + \sum_{k \in \{1..r, -\}} \sum_{j \in \{1..n, -\}} f_1(b'_j, c_k) w(b'_j, c_k) \\
&= w(f_1)
\end{aligned}$$

This proves the theorem. 2

Theorem 8. *If W has integer values, then $d_{X,d,M,W}^N(A, B)$ can be computed in polynomial time in $size_W(A)$ and $size_W(B)$.*

PROOF: The weights and capacities of the graph of the minimal weight maximal flow problem associated with this metric can be computed in $\#A \cdot \#B$ time. These numbers are all integers. The minimal weight maximal flow problem can be solved in polynomial time in $size_W(A)$ and $size_W(B)$ [10]. This proves the theorem. 2



Fig. 3 Network for $d^N(A, C)$

Example 2. Assume one has to choose among a number of 7 element sets the one most representative for a 100 element set B . Without weights, only the *best* 7 elements of B will determine the outcome. Using a weight 14 (or 15) for the elements in the 7-element sets, an element of the small set can match up to 14 (or 15) elements in B and 98 (or all) elements of B will influence the distance to a particular 7-element set. So one can expect a much better representative.

Corollary 1. d^m is a metric (theorem 2) and computable in polynomial time (theorem 3).

PROOF: This follows directly from proposition 2, theorem 7 and theorem 8 2

7 Applications

As mentioned earlier, metrics between point sets have applications in many areas. In this section we present some results from applications in machine learning.

7.1 The Diterpenes dataset

The diterpenes database (see [4]) describes 1503 diterpenes, which are organic compounds of low molecular weight with a skeleton of 20 carbon atoms. They are of significant chemical and commercial interest because of their use as lead compounds in the search for new pharmaceutical effectors. A common problem is to determine the structure of these diterpenes. Among other methods, one possibility is to get

information from NMR (nuclear magnetic resonance) spectra. The interpretation of these spectra normally requires specialists with detailed knowledge on peak patterns and chemical structures. Therefore, the use of machine learning methods to predict the structure of diterpenes could save costs.

Here, instance based learning will be used, which is a machine learning method that depends on the existence of a distance. The method takes as input a set of training examples of diterpene NMR spectra and the corresponding skeleton types (in this database, there are 23 different types). The type of new, unknown, diterpenes is predicted by comparing them to the set of known examples and predicting the same class as the nearest (most similar) training example. The nearest example is determined by measuring the distance between the unknown example and the training examples and choosing the smallest one.

The description of each diterpene consists of a set of peak frequencies together with their multiplicities. So, we are considering sets of points where each "point" is a tuple (frequency, multiplicity). For the distance between the points we use the euclidean metric.

The table below shows the results of our experiments on this dataset for some different instantiations of the schema given in section 3 and also the results obtained by other machine learning systems as reported in [4]. For all systems, the accuracy is based on a tenfold crossvalidation. This means the dataset is divided in ten parts. In each of the ten runs of the algorithm one of these parts is the test set and the other parts are the training set. The reported result is then the average of the test accuracy for all runs.

System	Accuracy
FOIL	78.3%
RIBL	91.2%
TILDE	90.4%
ICL	86.0 %
IBL - matchings	93.5%
IBL - linkings	85.0%
IBL - hausdorff	83.5%
IBL - surjections	84.4%
IBL - fair surjections	84.5%

As can be seen from the table, instance based learning using matching distances not only performs better than instance based learning with other distances but also better than other machine learning methods.

7.2 The Musk dataset

The musk dataset is described in [3]. This dataset describes a set of 102 molecules which are judged by human experts to be musks or non-musks. Each example is a set of 166-tuples. Each 166-tuple corresponds to one of the possible conformations of the molecule. We did a tenfold crossvalidation using different methods. We used an euclidian metric between the points (166-tuples). The following table summarizes the results. Also, the results of some other algorithms are included (see [3]).

iterated-discrim APR	92.4
all-positive APR	80.4
backpropagation	75.0
C4.5 (pruned)	68.5
IBL - matchings	88%
IBL - hausdorff	82%
IBL - linkings	79%
IBL - surjections	85%
IBL - fair surjections	49%

Again, our metric based on matchings gives the best results for instance based learning. Only Iterated-Discrim APR gives better results. The latter is an algorithm that searches axis parallel rectangle hypotheses, using the extra knowledge that an example is musk iff the 166-tuple of one of its conformations is in an (unknown) hyper-rectangle. C4.5, backpropagation and IBL did not make use of this information.

8 Summary

This paper studies the problem of extending a metric between points to a metric on the space of (finite) sets of points. A measure based on optimal matchings was proposed and was proven to be a metric. It was shown that this metric can be computed in polynomial time in the size of its arguments. Next, a normalised version of this metric, also computable in polynomial time, was proposed. Also, a variant was developed which associates weights with the points in a set. It is better suited to measure distances when the involved sets have very different cardinalities.

Finally, we have reported on two experiments in the machine learning area. They show that our metric gives substantially better results than the use of other similarity measures. Moreover, our results are better than those obtained with several learning algorithms based on

other principles. This shows that the proposed metric is not only of theoretic interest and supports the intuition that a good metric is preferable above a similarity measure.

Acknowledgements

Jan Ramon is supported by the Flemish Institute for the Promotion of Science and Technological Research in Industry (IWT). Part of this work was done while Maurice Bruynooghe was supported by the Fund of Scientific Research, Flanders. This work was also partly supported by the European community Esprit project no. 20237, Inductive Logic Programming 2 and by the GOA LP+ project. We are grateful to the reviewers for careful reading and several useful suggestions.

References

1. G. Bisson. Conceptual clustering in a first order logic representation. In *Proceedings of the Tenth European Conference on Artificial Intelligence*, pages 458–462. John Wiley & Sons, 1992.
2. H. Blockeel, L. De Raedt, and J. Ramon. Top-down induction of clustering trees. In *Proceedings of the 15th International Conference on Machine Learning*, pages 55–63, 1998. <http://www.cs.kuleuven.ac.be/~ml/PS/ML98-56.ps>.
3. T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez. Solving the multiple-instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1-2):31–71, 1997.
4. S. Džeroski, S. Schulze-Kremer, K. R. Heidtke, K. Siems, D. Wettschereck, and H. Blockeel. Diterpene structure elucidation from ^{13}C NMR spectra with inductive logic programming. *Applied Artificial Intelligence*, 12(5):363–384, July-August 1998.
5. T. Eiter and H. Mannila. Distance measures for point sets and their computation. *Acta Informatica*, 34, 1997.
6. W. Emde and D. Wettschereck. Relational instance based learning. In *Proceedings of the 1995 Workshop of the GI Special Interest Group on Machine Learning*, 1995.
7. W. Emde and D. Wettschereck. Relational instance-based learning. In L. Saitta, editor, *Proceedings of the Thirteenth International Conference on Machine Learning*, pages 122–130. Morgan Kaufmann, 1996.
8. Alt H., Mehlhorn K., Wagener H., and Welzl E. Congruence, similarity and symmetries of geometric objects. *Discrete Computational Geometry*, (3):237–256, 1998.
9. P. Langley. *Elements of Machine Learning*. Morgan Kaufmann, 1996.
10. K. Mehlhorn. *Graph algorithms and NP-completeness*, volume 2 of *Data structures and algorithms*. Springer, 1984.
11. J. Ramon and M. Bruynooghe. A framework for defining distances between first-order logic objects. In *Proceedings of the Eighth International Conference on Inductive Logic Programming*, Lecture Notes in Artificial Intelligence, pages 271–280. Springer-Verlag, 1998.

12. Jan Ramon and Maurice Bruynooghe. A polynomial time computable metric between point sets. Technical Report CW-301, Department of Computer Science, Katholieke Universiteit Leuven, October 2000. <http://www.cs.kuleuven.ac.be/publicaties/rapporten/CW2000.html>.
13. Grimaldi R.P. *Discrete and combinatorial mathematics*. Addison-Wesley, 1989.